



Logique du fonctionnement et logique de l'utilisation

J.F. Richard

► To cite this version:

J.F. Richard. Logique du fonctionnement et logique de l'utilisation. RR-0202, INRIA. 1983. inria-00076356

HAL Id: inria-00076356

<https://inria.hal.science/inria-00076356>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT NATIONAL DE RECHERCHE
CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP 105

78153 Le Chesnay Cedex
France
Tél (3) 954 90 20

Edl Dif

Rapports de Recherche

N° 202

LOGIQUE DU FONCTIONNEMENT ET LOGIQUE DE L'UTILISATION

Jean François RICHARD

Avril 1983

GROUPE DE PSYCHOLOGIE ERGONOMIQUE

RAPPORT DE RECHERCHE

LOGIQUE DU FONCTIONNEMENT ET LOGIQUE DE L'UTILISATION

J.F. RICHARD

Professeur à l'Université Paris VIII

Conseiller Scientifique à l'INRIA



PAPIER RECUPERE ET RECYCLE

SUMMARY

Understanding how an apparatus works and how to use it to complete a definitive task is analyzed within the framework of the distinction between declarative and procedural knowledge. A review of the psychological studies on learning how to use an information processing device is presented. The thesis is that when a subject knows how such a device works, he does not know necessarily how to proceed with it to complete a definite task : a deductive work is needed, which may be important. It is suggested to try to define utilization rules, describing procedures for completing macroactions using a set of commands, and being consistent with the representation the subject has about goals and sub-goals when planning his action.

Functioning rules, utilization rules, representation, declarative knowledge, procedural knowledge.

RESUME

Le problème de la compréhension du fonctionnement d'un dispositif et de l'élaboration de règles d'utilisation est situé dans le cadre de la distinction entre connaissance déclarative et connaissance procédurale. Une revue de la littérature sur les études psychologiques portant sur la compréhension du fonctionnement d'un dispositif est présentée. La thèse est que passer de la connaissance du fonctionnement à l'élaboration d'une procédure satisfaisant aux contraintes d'une tâche définie implique un travail déductif important, qui est loin d'être simple et automatique. L'approche suggérée est d'essayer de définir des règles d'utilisation relatives à des macro-actions, impliquant plusieurs commandes, qui soient compatibles avec la représentation que l'opérateur se donne concernant les buts et sous-buts de son action.

Règles de fonctionnement, règles d'utilisation, représentation, déclaratif, procédural.

Dans la présentation des connaissances nécessaires à l'utilisation d'un dispositif on a affaire à deux modes d'exposition. L'un met l'accent sur le fonctionnement du dispositif : on explique quel est l'effet de chaque commande, quelles sont les modifications qu'elle produit dans l'état du système. L'autre indique comment utiliser le système pour obtenir un résultat donné : dans ce cas on fournit en général des exemples d'utilisation et on donne pour chacun d'eux la suite des commandes ou instructions à utiliser.

Dans la lecture de notices d'utilisation on rencontre les deux types d'énoncés et l'on passe souvent d'exposés très généraux à des cas très particuliers d'utilisation. En général les informations données sur le fonctionnement sont insuffisantes pour que l'utilisateur soit à même de comprendre comment la suite des commandes produit le résultat obtenu et donc pourquoi cette suite est adéquate si l'on souhaite obtenir ce résultat. Par ailleurs, les exemples d'utilisation qui sont présentés sont trop fragmentaires pour que l'utilisateur ait une idée suffisamment complète des diverses utilisations possibles de l'appareil.

Cela peut expliquer pourquoi les usagers utilisent peu les notices et cherchent plutôt à "apprendre en faisant" plutôt que de commencer par lire attentivement les instructions (Szlichcinski, 1979; Lewis, Mack 1982; Wright 1981).

Notre hypothèse de travail est que ces difficultés tiennent à ce qu'on ne prend pas en compte le fait que le langage du fonctionnement et celui de l'utilisation se réfèrent à deux logiques très distinctes et que si l'on peut toujours en principe passer de la première à la seconde (puisque une utilisation donnée ne peut être justifiée que par le mode de fonctionnement), cela est loin d'être facile en fait et dans un certain nombre de cas cela présente des difficultés cognitives considérables.

Le problème peut se diviser en plusieurs questions :

- Quelles connaissances est-il suffisant de donner à l'utilisateur pour satisfaire ses besoins d'utilisateur ? La réponse à cette question n'est pas simple dans la mesure où l'utilisateur d'un appareil n'a pas besoin, pour ce qu'il doit faire, des mêmes connaissances qu'un dépanneur ou a

fortiori qu'un ingénieur-concepteur. Cette question a fait l'objet de recherches approfondies, celles qui concernent la notion de "modèle conceptuel", (Moran 1981, Young 1981), de "machine minimale" (Cuny, Deransart 1972).

- Comment l'opérateur passe-t-il de la connaissance du fonctionnement au mode d'utilisation ? Cette question, qui nous paraît centrale, n'a pas à notre connaissance été abordée de front. On sous-estime à notre avis les difficultés cognitives liées à ce passage, faute de connaissances sur les processus psychologiques en jeu. Une réponse satisfaisante à cette question conditionne les méthodes à mettre en oeuvre pour un meilleur apprentissage des règles d'utilisation.

Le problème des relations entre connaissance du fonctionnement et procédures d'utilisation est apparenté à un problème beaucoup plus général en psychologie : celui des relations entre les structures et leur application à des contenus particuliers, problème actuellement posé par l'école de Genève à partir de la distinction établie entre analyse structurale et analyse procédurale (Inhelder et al. 1976, Ackerman-Vallado 1977, 1980). La première définit l'ensemble des possibilités du sujet à un niveau donné de développement, la seconde permet de déterminer quelles sont celles qui sont effectivement mises en oeuvre dans une tâche précise. C'est une interprétation possible de la notion de décalage horizontal introduite par Piaget pour expliquer que ne soient pas acquis en même temps des invariants qui ont les mêmes propriétés structurales, (comme ceux de substance, poids, volume) mais concernent des contenus différents. Cela revient à dire que des mécanismes spécifiques sont nécessaires pour expliquer l'application de structures générales à des contenus particuliers. Fischbein (1978) parle à ce sujet de structures accommodatives. On peut citer également la distinction que fait Reuchlin (1973) entre processus de formalisation et processus de réalisation (voir aussi Lemercier 1982).

Les relations entre connaissances relatives à la structure et connaissances relatives à l'utilisation ont fait l'objet d'une réflexion approfondie en informatique. Cette réflexion porte sur le problème de la représentation des connaissances et de leur accessibilité dans un contexte particulier d'utilisation : cette problématique n'est pas d'origine psychologique mais elle pose le problème en des termes intéressants pour le psychologue.

1 - Le débat entre le point de vue déclaratif et le point de vue procédural en informatique

Winograd (1975) a donné une présentation assez synthétique des termes du débat. Il s'agit de savoir quelle est la façon la plus adéquate de représenter les connaissances. Dans une représentation sous forme déclarative on a des items de connaissance, les concepts, et des relations entre ces items, qui servent à les définir les uns par rapport aux autres. C'est le type de représentation adopté dans les réseaux sémantiques et qui peut convenir assez bien pour représenter la signification des mots du vocabulaire et les connaissances scientifiques. Dans ce genre de représentation on a un certain nombre de connaissances de base, spécifiques au domaine (les axiomes en mathématique, les lois générales en physique) et par ailleurs un ensemble de procédures très générales pour manipuler ces connaissances, au premier rang desquelles les règles de la logique.

Dans ce cas il est évidemment très économique de stocker les connaissances sous formes déclarative : quand on a besoin de les utiliser on met en jeu les processus de déduction et on dérive la forme appropriée sous laquelle la connaissance doit être formulée pour être utilisable dans le contexte particulier où on se trouve. Par exemple, je me demande si l'objet "x" a une propriété donnée "v" et je sais qu'il n'a pas la propriété "u" ; si j'ai parmi mes connaissances " $v \supset u$ " je peux déduire par un raisonnement de type modus tollens que l'objet n'a pas la propriété "v" : $(\neg u(x) \wedge v(x) \supset u(x)) \supset \neg v(x)$. La connaissance $v(x) \supset u(x)$ peut évidemment être utilisée dans beaucoup de déductions et il serait assurément moins économique d'avoir une représentation distincte pour chacune des ses utilisations.

En revanche, il y a d'autres cas où la connaissance que nous avons du monde est plus facilement exprimée sous forme de procédures que sous forme de relations entre les éléments qui en expriment la structure. Par exemple, ce que je sais du jeu d'échecs est plus facile à représenter sous forme de programmes d'actions (si la situation est comme ceci, plutôt faire ceci ...), que sous forme de propriétés relationnelles entre les éléments du jeu. Cela est également le cas quand nos connaissances sont très locales, en d'autres termes quand l'applicabilité des procédures dépend beaucoup du contexte. Dans ces cas-là il est difficile de trouver une forme déclarative équivalente et celle-ci serait vraisemblablement assez complexe.

En principe, on peut trouver une forme déclarative dont on puisse dériver les règles d'action correspondant à la représentation procédurale, mais cela peut être difficile et pas nécessairement économique. Beaucoup d'auteurs comme Winograd ont donné la préférence à une approche procédurale mais avec le développement des systèmes experts, on tend maintenant à privilégier un point de vue déclaratif (Laurière 1982). La raison est la suivante : quand la connaissance est exprimée sous forme procédurale, comme dans un programme Fortran par exemple, la structure de contrôle est totalement explicite, elle est entièrement définie par le programmeur de sorte que l'adjonction ou la suppression d'un élément dans la base de connaissances a des répercussions sur tout le programme qui doit être revu dans son ensemble. Dans le cas où l'écriture est de forme plus déclarative, comme dans les systèmes de production (Nguyen 1982), le contrôle de l'exécution des instructions est pris en charge par l'interpréteur sur la base des conditions énoncées dans la partie gauche des règles. Dans les systèmes de productions classiques l'aspect procédural est encore assez nettement marqué, dans la mesure où l'ordre des règles est important et permet de décider quelles règles sera appliquée, dans le cas où plusieurs sont applicables. Mais on peut imaginer (Laurière 1982) des systèmes de production dans lesquels il n'y a aucune règle de priorité préétablie. Dans ce cas la structure de contrôle est autonome par rapport au système des règles : elle est entièrement prise en charge par l'interpréteur. La conséquence importante est que ce type d'écriture est modulaire : il permet de changer des parties du programme (modification d'une ou de plusieurs règles), sans avoir à revoir l'ensemble de celui-ci.

La tendance actuelle en informatique est d'exprimer les connaissances sous la forme la plus générale possible et de laisser à l'interpréteur le soin de construire la procédure à élaborer dans les contextes particuliers d'utilisation.

Ce qu'on peut retenir de cette évolution c'est qu'on peut dissocier la structure de contrôle de la base de données et que partant de cette dernière le système est à même d'élaborer une procédure pourvu qu'on lui précise le but à atteindre. C'est ainsi qu'on s'achemine vers des programmes dans lesquelles le programmeur n'a pas à définir des procédures mais seulement des buts (Laurière 1982). La recherche des procédures pour atteindre un but déterminé sur la base des connaissances fournies au système est un travail purement déductif qui fait

appel aux procédures de déduction automatique qui ont fait leurs preuves dans la démonstration de théorèmes et la résolution de problèmes.

Il est donc en principe suffisant de fournir à l'utilisateur un modèle minimal consistant en un ensemble de connaissances sur le fonctionnement qui permette de dériver les procédures à mettre en oeuvre pour atteindre tel ou tel but.

2 - Le problème de la représentation du dispositif chez l'utilisateur

2.1 L'approche du modèle analogique

Comment former chez l'utilisateur une représentation adéquate ? Une approche du problème est de partir de l'idée développée par Ausubel (1963) que pour faire acquérir des connaissances il faut les connecter aux connaissances existantes et donc rechercher parmi ces dernières ce qui pourrait constituer un cadre, une structure de référence favorisant l'intégration des connaissances nouvelles, une sorte de schéma assimilateur pourrait-on dire. Dans cette ligne Mayer (1981) fournit à l'apprenti un modèle analogique grâce auquel les actions réalisées par la machine peuvent être mimées par des actions connues de l'individu.

Par exemple, pour faire comprendre une fonction de tri, on explique au sujet : imaginez un employé de bureau qui a trois casiers, un pointeur et une armoire à archives avec de nombreux casiers. Chaque tiroir contient un ensemble de cartes (cela correspond à un fichier pour l'ordinateur). Pour éviter de mélanger les cartes il sort toutes les cartes d'un tiroir à la fois et les met dans un des paniers placé à gauche sur son bureau (ce sont les données à traiter). Au milieu du bureau il y a un emplacement où l'employé met une carte à la fois. L'examen de la carte peut donner un résultat positif ou négatif (en fonction du critère du tri). L'un des paniers marqué "bon" sert à mettre les cartes pour lesquelles le résultat est positif, l'autre (marqué "rebut") sert à mettre les cartes pour lesquelles le résultat est négatif.

L'employé prend une carte sur la pile du panier de gauche, la place à l'emplacement test pour l'examiner, et la dépose sur le haut de la pile du panier "bon" si le résultat est positif ou sur le haut de la pile du panier "rebut" si le résultat est négatif. Il continue jusqu'à ce que toutes les cartes soient passées en revue.

Cette approche donne des résultats intéressants dans le cas de novices (Mayer 1981), mais elle peut présenter des inconvénients (Halasz, Moran 1982).

Dans une analogie, la correspondance n'est jamais totale : il y a toujours des aspects non pertinents mais l'apprenti ne sais pas lesquels. Ainsi, un fichier peut bien être considéré comme un tiroir dans lequel on range des cartes sur lesquelles il y a des informations. Dans un tiroir les cartes données sont bien rangées selon un certain ordre (comme les données dans un fichier), mais il y a en général d'autres indices de repérage, par exemple des indications alphabétiques ou des rubriques comme dans des fichiers de bibliothèque. S'il raisonne à partir du modèle analogique, l'apprenti risque de prendre en compte ces aspects non pertinents qui constituent alors des obstacles pour l'élaboration d'une procédure adéquate.

L'utilisation privilégiée de l'analogie est sans doute la référence au langage naturel et la désignation des commandes par des mots familiers. Il est sans doute illusoire de croire que la possibilité de dialoguer en langage naturel résout l'essentiel des problèmes de communication homme-machine (Plum 1977, cité dans Du Boulay et al. 1981). Scapin (1982) montre que contrairement à la croyance répandue, de nombreuses difficultés subsistent quand les commandes correspondent à des mots familiers et évocateurs pour le sujet. Ceux-ci ont en général plusieurs significations et il n'y a pas le contexte de la phrase ou du récit pour lever les ambiguïtés : chez les sujets non expérimentés les relations entre code et fonction s'établissent difficilement et restent instables.

2.2 L'approche du modèle de fonctionnement

Une autre approche du problème part de l'idée que les systèmes informatiques sont trop différents des systèmes familiers pour qu'il existe des modèles analogiques satisfaisants, et qu'il vaut mieux construire des modèles conceptuels adéquats et faire cette construction pièce par pièce plutôt que de rechercher une appréhension globale approchée.

Cuny et Deransart (1972a, 1972b), ont proposé un mode de description des dispositifs qui se réfère à ce point de vue. Décrire un système consiste à l'analyser à partir d'un nombre de variables discrètes (ou traits distinctifs) le plus petit possible mais suffisant pour représenter les différentes situations dans lesquelles l'opérateur doit intervenir et à formuler un certain

nombre de règles prenant en compte l'ensemble de ces situations. L'ensemble de ces règles constitue une grammaire. Les connaissances de l'opérateur peuvent également s'exprimer en termes de règles définissant pour chaque action ses conditions d'application et les modifications produites dans le système. La connaissance qu'a l'opérateur du système est représentée par le degré de correspondance entre les deux ensembles de règles.

Moran (1981) a proposé récemment un outil de description des systèmes informatiques fait du point de vue de l'utilisateur qui est une systématisation de cette approche : il s'agit d'une grammaire décrivant le format général d'un langage de commande et applicable à différents types de systèmes. Quatre niveaux sont distingués :

- le niveau de la tâche : il s'agit de l'analyse des tâches, correspondant aux objectifs de l'utilisateur, qui sont réalisables à l'aide du système.
- le niveau sémantique : il s'agit à ce niveau de décrire les objets composant le système et les opérations réalisables sur ces objets. C'est à l'aide des modifications produites dans le système par les opérations que doivent être réalisées les tâches du niveau 1.
- niveau syntaxique : il concerne la syntaxe des commandes (types de commande, arguments à spécifier, contexte d'utilisation). A chaque commande correspond une ou plusieurs opérations du niveau sémantique.
- niveau de l'interaction : les actions matérielles (frappe de touches ou autres types de manipulations), par lesquelles l'utilisateur communique avec le système.

Mayer et Bayman (1981) ont présenté une description du fonctionnement d'une calculatrice de poche proche de cette perspective et s'éloignant notablement de l'approche analogique décrite plus haut. L'effet des commandes est exprimée en termes de productions c'est-à-dire de règles ayant une partie condition et une partie action. L'action consiste en une suite d'opérations de base. Chaque opération de base consiste en une transformation opérée sur un objet en un lieu. Les transformations possibles sont : trouver un objet, en créer une copie, le détruire, en calculer la valeur. Les objets sont des nombres, des opérations, des symboles d'opérations arithmétiques, des expressions. Les lieux sont le registre d'affichage, le registre interne, le clavier.

Voici un exemple de commande :

Si on appuie sur la touche "+" après avoir appuyé sur une touche chiffre alors la valeur affichée ne change pas et le registre interne contient une expression formée de la valeur numérique de l'expression qui était dans le registre interne et du signe "+". Ce qui se passe dans le registre interne est détaillé par la suite des opérations de base suivantes :

- trouver l'expression qui est dans le registre interne.
- trouver la valeur numérique.
- effacer l'expression du registre interne .
- placer dans le registre interne la valeur de l'expression suivie du signe "+".

La préoccupation des auteurs est en fait d'exprimer la représentation que se font les sujets du fonctionnement de la calculette dans les mêmes termes, c'est-à-dire sous forme de productions. Pour juger de l'adéquation de la représentation des sujets il suffit de comparer le système de production qui caractérise chacun d'eux avec le système qui représente le modèle de l'appareil.

Les modes de description développés par ces différents auteurs sont, on le voit, très voisins. Il s'agit toujours de trouver une description de la machine en termes d'automate, d'exprimer les modifications de l'état de l'automate (sur l'ensemble des variables entrant dans la description, pour chaque manipulation réalisable avec l'appareil, dans les différents contextes possibles). En fait il est loin d'être facile de trouver un ensemble de règles permettant de simuler le comportement d'une machine, même dans le cas d'un appareil apparemment simple comme une calculette.

Tout d'abord ce comportement peut être très complexe. Young (1981), montre que s'il y a un modèle simple pour les calculettes à logique polonaise inversée, n'est pas le cas pour les autres modèles de calculettes.

En second lieu il s'agit de bien choisir le niveau de finesse de la description. Cette description n'est pas faite, on l'a compris, au niveau des constituants matériels de l'appareil. Cette description est faite du point de

vue de l'utilisateur et donc du point de vue des actions qu'il est susceptible de faire sur la machine. Elle doit fournir une logique susceptible d'expliquer les effets visibles des manipulations et pour cela elle doit faire intervenir des effets invisibles, qui n'ont pas nécessairement une réalité matérielle mais qui donnent une cohérence aux effets visibles. Ce sont au sens le plus précis du terme, des modèles de simulation.

L'objectif de ces modèles est double : permettre d'élaborer à partir des commandes disponibles des procédures satisfaisantes pour réaliser les tâches que se propose l'opérateur (c'est-à-dire adéquates et économiques), comprendre ce qui s'est passé en cas d'incident, de fausse manœuvre, afin de pouvoir y remédier.

Un modèle adéquat pour un type de tâche n'est pas nécessairement le meilleur pour un autre type de tâche : des choix différents peuvent être faits suivant que la tâche envisagée est l'exécution de procédures routinières, l'invention de nouvelles procédures ou encore l'évaluation de procédures et la correction d'erreurs.

On peut regarder le fonctionnement de la machine de deux points de vue comme le souligne Young (1981), qui distingue deux types de modèles de fonctionnement. L'un est un modèle de registre qui donne une description de la machine en termes de changements d'état des registres visibles ou invisibles de la machine et permet de comprendre (et de prévoir) le comportement de la machine en réponse à une séquence arbitraire de commandes n'ayant aucune finalité assignable. Le second modèle consiste à représenter le fonctionnement de telle manière qu'on puisse établir une correspondance entre ce que fait la machine, la tâche du sujet (le calcul), telle qu'on peut la décrire indépendamment de la machine, et les actions à réaliser avec la machine pour accomplir la tâche à l'aide de celle-ci. Cela signifie trouver trois descriptions concernant respectivement les changements d'état de la machine, la tâche et les actions, qui soient telles qu'à chaque opération élémentaire dans une description, on puisse faire correspondre une opération dans les deux autres.

Dans le premier cas, dit Young, on s'intéresse à la mécanique de la machine, dans le second à son utilisation. En fait il serait plus juste de dire que le premier cas correspond également à une forme d'utilisation, moins fréquente

certes quand il s'agit de calculette, mais pas négligeable pour autant : la recherche et la correction de l'erreur dans le cas où on observe une discordance entre le résultat attendu et le résultat obtenu. Prenons le cas d'une erreur assez insidieuse que nous avons observée chez un étudiant. La calculette utilisée était un modèle algébrique qui suit, pour l'exécution des opérations, les règles de priorité de l'algèbre (priorité de la multiplication sur l'addition, parenthèses). L'étudiant voulait faire une somme des valeurs, l'élever au carré et diviser ce carré par le nombre de valeurs. L'étudiant a terminé la somme par la touche $+$ au lieu d'utiliser la touche $=$, qui est recommandée : il avait constaté en effet que taper $+$ ou taper $=$ avait le même effet, ce qui est vrai dans le contexte où on ne poursuit pas le calcul. Dans le cas, précis, taper $+$ au lieu de $=$ fait que la somme est ajoutée au carré de la somme dans le numérateur de la division au moment où on termine le calcul par la touche $=$. Il est difficile d'expliquer dans quels contextes $+$ a le même effet que $=$ et dans quels contextes ce n'est pas le cas, sans un modèle faisant intervenir des changements d'état de registre ; il faut une connaissance minimale du fonctionnement pour comprendre que l'utilisation de la touche $+$, pour exécuter un calcul, est appropriée dans certains cas, inappropriée dans d'autres.

Se plaçant dans une optique d'utilisation qui est en fait la recherche de procédures de calcul, Young examine quels sont les modèles d'utilisation adaptés pour les trois grand types de calculettes qui existent actuellement sur le marché : à notation polonaise inversée, type algébrique et version simple à 4 fonctions, sans parenthèse et sans règle de priorité. Il montre que les modèles à notation polonaise inversée conviennent (au sens de "il existe une correspondance simple entre les processus distingués plus haut") pour les calculs qui se représentent le plus commodément par un graphe. Par exemple diviser 3 par 4, ajouter 5 au résultat, élever au carré, etc. Si on doit évaluer une formule, il faut la mettre préalablement sous forme d'un graphe du type de celui représenté page 39. En revanche, l'évaluation d'une formule est la forme la plus adaptée pour les modèles algébriques : si on a un calcul représenté par un graphe, il faut d'abord le linéariser pour avoir une formule. Quant à la calculette à 4 fonctions, le modèle d'utilisation le plus simple est l'opération arithmétique de base avec un opérateur et deux opérandes. Dans ce cas la correspondance entre les processus peut être présentée ainsi :

: Processus de la tâche: Processus de la manipulation : Processus de la machine :		
: Premier opérande	: Entrer le premier nombre	: Stockage du premier
:	:	: argument dans le
:	:	: registre
: Opérateur	: Frapper touche opération	: Stockage de la fonc-
:	:	: tion dans le registre
: Second opérande	: Entrer le deuxième nombre	: Stockage du second
:	:	: argument dans le
:	:	: registre
: Evaluation	: Frapper la touche =	: Exécution du calcul et
:	:	: passage dans le registre:
:	:	: du premier argument (en
:	:	: vue de l'opération sui-
:	:	: vante).

Cette perspective est intéressante dans la mesure où elle permet de montrer qu'un type d'appareil ne permet pas nécessairement de concevoir un modèle simple de fonctionnement pour les diverses utilisations qui peuvent en être faites (et donc susceptible d'être efficient). Mais elle semble surtout pertinente du point de vue de la conception de dispositif. Elle montre en effet que le concepteur d'un système doit se préoccuper du modèle de la machine pour l'utilisateur et faire des choix techniques qui permettent de concevoir des modèles simples d'utilisation. Ce point de vue est d'ailleurs explicitement exposé par Moran : "The purpose of CLG (Command Language Grammar) is to allow (indeed force) the system designer to create a conceptual model of the system for the user to assimilate" (1981, p. 42).

Il reste qu'à notre avis prendre le problème sous le seul biais de la correspondance entre les modèles conceptuels permis par la machine et les procédures usuelles de l'utilisateur c'est n'en voir qu'un aspect. C'est ne se préoccuper que de la transposition des procédures connues et laisser de côté la possibilité d'apprentissage de nouvelles procédures, qui peuvent être plus efficaces que les procédures usuelles. Il faut aussi se demander quelles sont les conditions d'apprentissage du mode fonctionnement de la machine et aborder les études de nature plus directement psychologiques qui ont étudié l'acquisition du modèle par les utilisateurs.

3 - L'acquisition par l'opérateur de la représentation du dispositif

3.1 L'apprentissage à partir de la présentation d'un modèle de fonctionnement

Une première technique d'apprentissage consiste à utiliser l'analogie du fonctionnement avec des modèles extérieurs. Mayer (1981) a utilisé, pour faire comprendre des commandes d'un langage de gestion, l'analogie présentée précédemment (p. 7). Le groupe expérimental, qui dispose, outre du manuel, de ce commentaire supplémentaire se révèle supérieur au groupe contrôle dans des exercices qui ne sont pas la simple réplique des exemples fournis dans le manuel mais comportent des questions impliquant une organisation différente des instructions apprises. Toutefois, à notre avis, les différences n'apparaissent pas considérables, compte tenu du fait que le groupe expérimental dispose d'une information plus abondante et consacre vraisemblablement plus de temps à l'apprentissage.

Une seconde approche consiste à rendre visibles les changements d'état du système. Cela revient à construire une représentation matérielle du modèle conceptuel consistant non pas à reproduire les constituants matériels mais à donner une image du modèle simulant le comportement de la machine. Il s'agit bien d'une analogie en un certain sens puisque le modèle représenté n'est pas le modèle matériel mais il ne s'agit pas d'une référence à un modèle externe ayant une signification en dehors du système.

En vue de l'initiation à l'informatique, Du Boulay, O'Shea et Monk (1981), ont utilisé un dispositif comportant seulement 9 instructions :

Load, Store, Add, Decrement, Jump, Jump if zero, Input, Call, Or.

Un ensemble de 10 boutons situés sur la droite du clavier permet de les introduire en machine. Sur la gauche figure un autre ensemble de touches constituant un langage de commande pour les différentes fonctions : entrer les instructions, relire le programme, l'exécuter (en bloc ou pas à pas), consulter l'état des registres. La machine contient 8 emplacements-mémoire pour les données. Un registre d'affichage permet de consulter à tout moment l'état des différents registres de la machine. Une représentation analogue a été utilisée pour l'apprentissage du langage LOGO. Les auteurs ne présentent pas de données comparatives permettant de juger de l'effet de ce mode de présentation.

Mayer (1981) a également utilisé un modèle concret de l'ordinateur pour l'apprentissage d'instructions du type de celles du Basic. Il exploite au maximum les analogies avec d'autres dispositifs pour représenter les fonctions de base de la machine. En cela la méthode est voisine de l'approche analogique avec cette différence cependant que la référence au modèle analogique, ne vaut pas pour l'ensemble du système mais pour chacune de ces fonctions.

Le dispositif comporte une fenêtre permettant la lecture d'une carte à la fois à l'entrée des données et une fenêtre pour la sortie des données. La mémoire est figurée par une ardoise divisée en 8 cases ayant chacune une étiquette et pouvant contenir un seul nombre. On peut modifier le contenu d'une case en l'effaçant et en inscrivant une nouvelle valeur. Le programme est présenté sous la forme d'une liste : à chaque ligne est écrite une instruction. Cette liste passe devant un pointeur : c'est l'instruction qui est devant le pointeur qui est exécutée. Une fois celle-ci exécutée, la feuille avance d'un pas et le pointeur se trouve en face de la ligne suivante.

Les sujets du groupe expérimental disposent d'une feuille décrivant ce modèle en plus d'un manuel décrivant d'autres instructions du type : lire, écrire, assigner une valeur à une variable, évaluer une expression, tester l'égalité, aller à ... Le groupe contrôle n'a que le manuel. Le test comporte deux types d'exercices : écrire de petits programmes (de 1 à 5 instructions), décrire ce que fait une liste d'instructions. Dans une seconde expérience on présente au groupe expérimental le modèle avant la lecture du manuel, ce qui permet aux sujets de l'utiliser comme organisateur de l'information et devrait être bénéfique d'après la théorie d'Ausubel. Le groupe contrôle en revanche n'a communication du modèle concret qu'après la lecture du manuel. Cela ne devrait pas améliorer

rer l'apprentissage mais permet d'égaliser la quantité d'information fournie aux deux groupes.

Dans la première expérience le groupe expérimental a une meilleure performance que le groupe contrôle mais seulement pour les exercices comportant un plus grand nombre d'instructions (l'interaction est significative au seuil .05). Dans la seconde expérience les résultats vont dans le même sens mais la différence en faveur du groupe expérimental aux exercices complexes est atténuée; il y a en revanche une différence assez marquée en faveur du groupe expérimental aux exercices simples. Ce sont ces deux effets qui sont traduits dans l'interaction (significative à .05) et pas seulement le premier, de sorte que ces résultats n'apportent pas une confirmation indiscutable à l'hypothèse qu'un modèle concret améliore vraiment la compréhension. Le fait que l'avantage du groupe expérimental soit réduit dans la seconde expérience laisse penser qu'une partie au moins de l'effet constaté dans la première expérience tient au fait qu'il disposait de plus d'information que le groupe contrôle.

Récemment, More, Rancoule et Lamouroux (sous presse) ont exposé une méthode de simulation utilisée dans l'initiation à l'informatique et très voisine de celles décrites plus haut, mais qui a le mérite supplémentaire de présenter graphiquement sur écran, à la demande, ce que fait l'ordinateur. La machine simplifiée comprend un ensemble d'instructions de base très voisin de celui utilisé par Du Boulay, O'Shea et Monk (1981). L'originalité est la présentation graphique faite à la demande. Supposons, par exemple, que l'instruction à l'adresse 047 a comme contenu AJ 130 ce qui veut dire : additionner la valeur figurant à l'adresse 130 au contenu de l'accumulateur. On peut voir sur l'écran à gauche l'accumulateur avec son contenu, à droite en haut l'adresse 047 avec son contenu en bas à droite l'adresse 130 avec son contenu et les commentaires appropriés. Ensuite, on voit se dessiner des flèches allant du contenu de l'adresse 130 au contenu de l'accumulateur avec le signe "+". Puis l'exécution s'effectue et on voit le contenu de l'accumulateur se modifier par l'addition du contenu de l'adresse 130, tandis que ce dernier reste inchangé. La présentation semble attrayante mais il n'y a pas encore de données empiriques permettant de juger de l'efficacité de la technique : les auteurs se proposent de prolonger dans ce sens leur étude.

Dans une recherche toute récente Friemel, Richard et Weil-Barais (1982b) ont simulé le fonctionnement d'une calculatrice le modèle T 130 de Texas, qui fonctionne suivant la logique algébrique. On a élaboré un modèle de fonctionnement de cette machine qui permet de simuler son fonctionnement pour les séquences de commandes constituant une procédure acceptable, (à l'exception de cas particuliers comme l'utilisation d'une constante de calcul) ou susceptibles de recevoir une interprétation ou encore correspondant aux erreurs rencontrées chez les sujets. On a considéré que les autres cas représentaient des manipulations non intentionnelles ou ininterprétables et dans ces cas le simulateur envoie un message d'erreur.

Le modèle de la machine comporte trois registres : le registre d'affichage, le seul visible sur la machine, le registre mémoire (une case) et un registre que nous appellerons le registre de travail : il correspond à peu près au registre interne de Mayer (1981b). Dans ce registre sont stockées les opérands et opérateurs des opérations non exécutées.

Voici un exemple donnant une idée du fonctionnement du modèle :

Commandes	Affichage	Mémoire	Registre de Travail
2	2	0	-
+	2	0	2 +
3	3	0	2 +
x^2	9	0	2 +
x	9	0	2 + 9 x
4	4	0	2 + 9 x
0	40	0	2 + 9 x
=	362	0	-
STO	362	362	-
5	5	362	-
SUM	5	367	-
8	8	367	-
÷	8	367	8
RCL	367	367	8
=	.02179837	367	-

Le registre de travail est modifié par l'introduction d'un opérateur. Au moment de l'exécution le second opérande de l'opération qui sera exécutée en premier se trouve dans le registre d'affichage. La simulation de ce modèle de la calculette est réalisée sur un microordinateur APPLE II. Les touches du clavier ont été réassignées aux diverses commandes de la calculette. Au centre de l'écran est dessiné un cadre figurant le registre d'affichage. En haut à droite un second cadre représente le registre mémoire, en haut à gauche un troisième cadre représente le registre de travail. A chaque manipulation de touche apparaissent sur l'écran les modifications de chaque registre. Lorsque l'exécution de plusieurs opérations est déclenchée, l'exécution se passe au ralenti de sorte qu'on peut voir les résultats partiels. Ainsi, dans l'exemple ci-dessus on verrait successivement $2 + 360$ et 362 .

Les sujets ont à leur disposition un manuel ressemblant à la notice distribuée avec la calculette par le constructeur et reçoivent quelques explications supplémentaires concernant la nature des registres. Ils sont par groupes de 2 ou 3 autour du microordinateur. Dans le manuel figurent des exemples de calcul à réaliser à l'aide de l'appareil. Les sujets du groupe contrôle travaillent individuellement avec la calculette. L'apprentissage dure de 30 à 45 minutes selon les sujets, qui sont des étudiants de psychologie de première année suivant un cours de statistiques.

Le test de l'apprentissage comporte deux parties : au cours de la première les sujets doivent découvrir une procédure économique pour calculer la variance d'un protocole en utilisant la formule :

$$\frac{\sum_{i=1}^n x^2 i - \frac{\left(\sum_{i=1}^n x i \right)^2}{n}}{n-1}$$

ou n est le nombre d'observations x la valeur de chaque observation. La procédure la plus économique consiste à faire simultanément le calcul de la somme des x et celui de la somme des carrés des x . Pour cela il faut, après avoir entré chaque valeur sauf la dernière, utiliser les commandes suivantes SUM x^2 +,

et pour la dernière valeur faire $\boxed{-}$ au lieu de $\boxed{+}$. Si les sujets n'ont pas découvert cette procédure on leur indique qu'ils doivent introduire les données une seule fois en machine.

La seconde partie du test consiste à prévoir, pour une suite de commandes, quel sera, après la frappe de la touche correspondant à chacune d'elles, l'état du registre d'affichage et l'état du registre mémoire.

L'hypothèse de travail était que les sujets du groupe expérimental auraient une meilleure représentation du fonctionnement de la machine, ce qui devrait se traduire par des prédictions plus exactes sur l'état des registres dans la dernière partie de l'épreuve, et que cette meilleure représentation faciliterait la découverte de la procédure optimale dans le calcul de la variance. Ni l'une ni l'autre de ces hypothèses n'ont été confirmées. Les performances des deux groupes sont les mêmes dans le problème et dans les deux parties de l'épreuve-test. Certes, on peut remarquer que les conditions d'apprentissage n'étaient pas strictement les mêmes dans les deux groupes : les sujets du groupe contrôle travaillaient individuellement alors que ceux du groupe expérimental travaillaient à deux ou trois et il est possible que l'efficacité de l'apprentissage en ait souffert. Il reste que les sujets n'ont pas profité de l'information fournie à l'écran sur ce qui se passait dans la machine à la suite de chaque commande.

Les expérimentateurs ont remarqué que les sujets ne prêtaient pas beaucoup attention à ce qui se passait dans le registre de travail mais regardaient surtout ce qui se passait dans le registre d'affichage : ils semblaient plus préoccupés par les résultats du calcul et la vérification de l'identité du résultat obtenu et de celui attendu, qui figurait sur la notice. Il est certain que la notice d'utilisation insiste surtout sur la façon de procéder pour faire les calculs et très peu sur le fonctionnement de la machine. Si les sujets avaient eu pendant l'apprentissage des exercices consistant à prédire l'état du registre d'affichage et de la mémoire après chaque commande comme dans le test, les résultats auraient peut-être été différents. Ce dernier type de tâche en effet focalise vraisemblablement plus l'attention sur le fonctionnement.

On peut remarquer, toutefois, que ces exercices correspondent davantage aux tâches que les sujets se proposent habituellement de réaliser avec une calculatrice que les exercices proposés dans le test. On peut faire l'hypothèse que si les sujets ont peu prêté attention aux informations sur le fonctionnement c'est qu'en fait ces informations ne correspondaient pas à leurs objectifs, qui étaient d'obtenir un résultat et d'apprendre comment faire pour l'obtenir, plutôt que de comprendre comment la machine fait pour l'obtenir. Leur objectif serait dans ces conditions de réussir plutôt que de comprendre (Piaget 1974). Ceci pourrait expliquer l'observation souvent faite (Szlichcinski 1979, Lewis et Mack 1981), que les sujets négligent souvent de lire les instructions figurant dans les notices et préfèrent procéder par essais et erreurs pour obtenir le résultat cherché. C'est seulement en cas d'échec qu'ils se reportent aux explications. Vermersch (1976) a observé également, dans une expérience d'apprentissage d'utilisation de l'oscilloscope, que les sujets ayant eu préalablement un cours théorique et disposant d'une notice d'utilisation manipulent au début les boutons d'une façon qui est liée non pas à leurs propriétés fonctionnelles mais à des caractéristiques spatiales (comme la proximité sur l'appareil), de sorte que ce comportement a l'aspect d'un tâtonnement qui n'est pas dirigé par les connaissances apprises. La recherche ne devient organisée qu'après une phase d'apprentissage impliquant la manipulation du dispositif.

En tout cas, même si l'idée que d'un point de vue pédagogique il est préférable de présenter de façon visible le fonctionnement de la machine, est très attrayante, si attrayante même qu'on est porté à penser qu'il n'est pas besoin d'une vérification empirique pour fonder sa validité, on ne peut actuellement tenir pour démontré que cela suffit pour assurer un apprentissage efficace. Les données publiées ne sont pas nombreuses et celles qui existent ne sont pas décisives. D'autres recherches sont nécessaires avant de pouvoir formuler une conclusion plus définitive.

3.2 L'apprentissage par l'adaptation aux contraintes de fonctionnement du dispositif

Il nous paraît intéressant d'investiguer l'hypothèse que lorsqu'un sujet apprend à utiliser un appareil, son objectif est de trouver une procédure pour réaliser la tâche et que ce qui l'intéresse c'est d'abord d'obtenir un résultat, ce n'est pas de comprendre. Ce n'est que s'il est impossible de réussir sans comprendre, qu'il s'intéresse au fonctionnement.

Les situations de programmation sont celles pour lesquelles il est le plus important de comprendre pour réussir et dans lesquelles en principe on doit être le plus à même de mettre en évidence un effet favorable de la transparence du fonctionnement. Une autre hypothèse est que les règles de fonctionnement ne sont apprises qu'à partir d'un apprentissage où il faut élaborer une procédure compatible avec ces règles, prendre conscience de l'inadéquation des procédures habituelles et inférer les propriétés critiques du fonctionnement à partir des résultats des tentatives infructueuses. Une expérience de Hoc (1979, 1980, 1981) nous paraît être une bonne illustration de ce propos.

Le dispositif utilisé permet d'étudier l'acquisition des règles de fonctionnement qui régissent le mode de lecture et le mode de sortie des données dans le cas de fichiers séquentiels. La situation choisie concerne la mise à jour de stocks. Un fichier (AS) indique l'état de l'ancien stock : c'est un tableau de 2 colonnes : la première indique par un nombre le numéro des pièces qui sont rangées dans l'ordre, la seconde la quantité en magasin. Un fichier (MVT) indique les mouvements de pièces effectués dans la journée. Ce fichier comporte aussi 2 colonnes : la première indique le numéro de la pièce impliquée dans le mouvement, la seconde indique la modification (nombre avec signe + pour une entrée, nombre avec signe - pour une sortie). Une pièce qui n'a pas eu de mouvement ne figure pas dans le tableau, une pièce qui a eu plusieurs mouvements sont rangées dans l'ordre. La tâche du sujet est de remplir le tableau nouveau stock (NS).

Le dispositif est le suivant :

- Il y a 2 cases-mémoires.
 - L'une (case A) peut recevoir une ligne du tableau ancien stock et l'autre (case M) une ligne du tableau mouvement.
 - Des touches permettent de commander l'exécution des instructions.
- Ce sont (pour celles qui nous intéressent dans cette présentation):

ENTRER A : La première ligne du tableau s'inscrit dans la case A. Elle disparaît du tableau et toutes les lignes remontent d'un cran.

ENTRER M : Même effet que la précédente mais pour le tableau mouvement.

ADDITIONNER : Addition des nombres en A et M et résultat en A.

COPIER : Copier le contenu de A dans le fichier nouveau tableau.

TEST A = M ? : La réponse du test est OUI si les numéros dans A et M sont les mêmes et NON s'ils sont différents.

Dans une première phase sont visibles les tableaux des données (AS et MVT) et des résultats (NS) et le contenu des cases-mémoires A et M; le sujet peut utiliser les premières commandes mais non le test. Les données montrent que l'apprentissage se fait par transfert et adaptation d'une procédure connue.

Le comportement des sujets correspond à trois règles :

- si la pièce n'a pas de mouvement alors : ENTRER A - COPIER,
- si la pièce a un mouvement : ENTRER A - ENTRER M - ADDITIONNER - COPIER
- si la pièce a plus d'un mouvement : ENTRER A - ENTRER M - ADDITIONNER - ENTRER M - ADDITIONNER ENTRER M ADDITIONNER COPIER.

La classification des pièces n'est pas modifiée par rapport à la procédure exécutable à la main dans laquelle on distingue trois modes de traitement. S'il n'y a pas de mouvement recopier l'état du stock ; s'il y a eu un mouvement additionner (ou soustraire) la quantité impliquée dans le mouvement, s'il y a eu plusieurs mouvements faire le cumul des mouvements successifs et ajouter le résultat à la valeur de l'ancien stock. Il faut par contre adapter la façon de procéder en chaque cas aux opérations permises par le dispositif : on ne peut copier que ce qui est en A, il faut donc le mettre en A, si ce n'est pas déjà en A ; on ne peut additionner que le contenu de A et de M, il faut donc mettre en A et M les contenus à additionner et s'il y a plusieurs mouvements il faut faire une addition cumulée. Pour chacun des mouvements suivants il faut faire : ENTRER M - ADDITIONNER et pour le dernier terminer par COPIER. Cette adaptation ne concerne pas la structure des règles mais la modification de chacune des règles : le découpage de l'action n'est pas modifié il y a toujours trois cas à distinguer et ce sont les mêmes. La règle acquise d'abord est celle qui concerne les pièces à un mouvement, et à partir de là sont construites par différenciation les règles pour les cas sans mouvement et ceux à plusieurs mouvements.

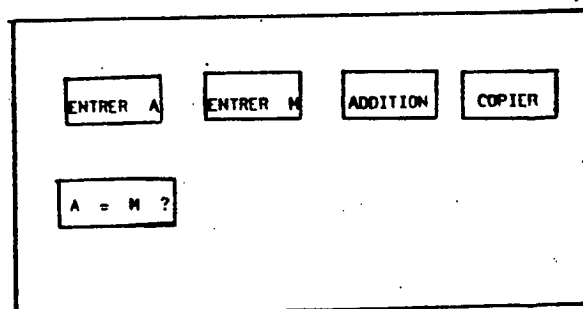
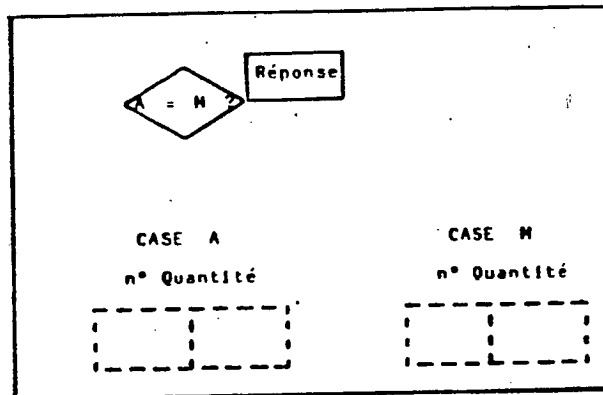
[illegible]

Figure 1 : Schéma du dispositif de l'expression de J.M. MOC en traits pleins ce qui est visible pendant les 3 phases en tirets : ce qui n'est visible que dans les phases 1 et 2 en pointillés : ce qui n'est visible qu'en phase 1.

Dans une seconde phase ne sont visibles que les contenus des cases mémoire. Le sujet ne disposant plus des tableaux ne peut plus savoir combien il y a de mouvements pour chaque pièce. Ce que lui permet de faire le dispositif c'est seulement de vérifier si les numéros en A et M sont les mêmes ou non, d'où l'on peut déduire si la mise à jour est terminée pour cette pièce ou non. En principe il y a donc deux choses à faire dans cette situation : découvrir d'abord quels sont les cas à considérer et trouver la suite des commandes à appliquer dans chaque cas. Comme le test a 2 issues, le plus simple logiquement est d'envisager 2 cas :

- les numéros sont différents donc la mise à jour est terminée.
- les numéros sont les mêmes donc la mise à jour n'est pas terminée.

Si la mise à jour est terminée faire COPIER puis ENTRER A et ENTRER M (puisque l'on a besoin d'avoir en mémoire la pièce suivante et le mouvement suivant pour pouvoir comparer les numéros). Si la mise à jour n'est pas terminée faire : ADDITIONNER (pour la mise à jour) et ENTRER M (pour faire la comparaison). Il convient donc ici de remodeler complètement l'action en l'organisant de façon différente : il convient non seulement de faire des aménagements à l'intérieur des règles mais de modifier le système des règles en définissant de nouveaux sous-buts. Cette phase se révèle de fait beaucoup plus difficile pour les sujets : une erreur très persistante consiste à commander COPIER après ADDITIONNER au lieu de faire ENTRER M, qui reste associé à ADDITIONNER. Ces séquences qui correspondent à l'organisation de l'action dans la phase 1 constituent des unités difficiles à dissocier.

Dans la troisième phase le sujet ne voit plus le contenu des mémoires mais il dispose du TEST $A = M$? indiquant si les numéros en A et M sont les mêmes ou non (1). En principe, le sujet peut conserver l'organisation de l'action décrite ci-dessus pour la phase 2 :

(1) Dans cette phase le sujet dispose en fait d'un autre test, qui permet l'arrêt de la procédure. Nous n'en parlons pas ici, car il n'est pas pertinent pour notre propos.

- voir si la mise à jour est terminée à l'aide de TEST A = M.
- si oui faire COPIER, ENTRER A, ENTRER M.
- si non faire ADDITIONNER, ENTRER M.

La seule modification à faire est de remplacer par TEST A = M ? La comparaison des numéros faite par le sujet lui-même. On peut donc faire l'hypothèse que dans la phase 3 sera transférée la structure du processus de commande de la phase 2. En réalité, l'examen des erreurs montre que le sujet revient au processus de commande utilisé en phase 1. Ainsi, le test n'est pas utilisé d'emblée pour comparer les contenus des mémoires, opération que le sujet faisait lui-même en phase 2, mais est utilisé comme une comparaison entre le numéro figurant en case A de la mémoire et le numéro figurant sur la première des lignes du tableau Mouvement non encore introduites en mémoire, comparaison que le sujet faisait en phase 1 pour savoir combien la pièce avait de mouvements. Par ailleurs réapparaissent les mêmes erreurs qu'en phase 2 lesquelles correspondent à un transfert du processus de commande de la phase 1, par exemple faire COPIER après ADDITIONNER sans vérifier si la pièce n'a pas d'autre mouvement, ce qui permet de penser que le sujet fait l'hypothèse que la pièce n'a qu'un mouvement.

S'il en est ainsi, on peut se demander si en phase 2 les sujets ont bien appris le processus de commande que nous avons décrit et consistant à considérer qu'il y a deux cas à envisager, les cas où il y a une mise à jour à faire et ceux où il n'y en a pas. Une autre interprétation possible du comportement des sujets en phase 2 est la suivante : le sujet traite chaque pièce successivement comme dans la phase 1 et pour chaque pièce il se pose deux questions. La première question consiste à se demander si la pièce a un mouvement ou non et, pour avoir la réponse, à vérifier si les numéros sont identiques ou non. Si la réponse est négative le sujet fait "COPIER" et passe à la pièce suivante en faisant : "ENTRER A," "ENTRER M". Si la réponse est positive le sujet traite le mouvement (commande ADDITIONNER) puis se demande s'il y a encore des mouvements pour cette pièce : pour cela il fait "ENTRER M", et vérifie si les numéros en A et M sont les mêmes. Si la réponse est négative il fait "COPIER" et passe à la pièce suivante, si la réponse est oui, il traite le mouvement et se demande à nouveau s'il y a encore des mouvements pour cette pièce. Par ailleurs, le sujet a tendance à anticiper une réponse positive à la première question en supposant que la pièce a un mouvement et il peut arriver qu'en cas de réponse positive à la

première question il fasse "ADDITIONNER" et ensuite "COPIER" au lieu de "ENTRER M", autrement dit qu'il oublie de poser la seconde question (erreur typique de la phase 2).

Ainsi, dans la seconde phase, comme dans la première, l'action est organisée autour du traitement d'une pièce et le sujet répète cette procédure pour chaque pièce. Ce qui est conservé encore, c'est la partition trichotomique des cas à considérer: aucun mouvement, un mouvement, plus d'un mouvement. Ce qui est modifié, c'est la procédure d'identification du cas dans lequel on se trouve : l'identification ne se fait plus à partir du tableau mais de la mémoire et comme on ne peut avoir qu'une ligne du tableau en mémoire, l'identification requiert plusieurs comparaisons successives au lieu du balayage des lignes du tableau.

Dans la troisième phase le sujet ne peut plus faire un traitement pièce par pièce. Il ne sait pas le numéro de la pièce qui est en A, puisqu'il ne voit plus le contenu des cases A et B : il sait seulement par le résultat du test si les numéros sont identiques ou non. La partition des cas à considérer doit être nécessairement dichotomique : il faut pour cela ne plus se préoccuper de l'identité de la pièce qui est en A et formuler une seule question "le mouvement figurant en M concerne-t-il la pièce qui est en A ?" Les deux questions précédentes: "la pièce en cours de traitement a-t-elle un mouvement ?" et "a-t-elle encore un mouvement ?" doivent être abandonnées. Autrement dit, le traitement doit être centré sur les mouvements non sur les pièces.

La procédure devient alors :

- 1 - voir si le mouvement qui est en M concerne la pièce qui est en A (A = M ?).
- 2 - si c'est NON, mettre dans le tableau nouveau stock les données concernant la pièce qui est en A (COPIER) ; introduire en mémoire la pièce suivante (ENTRER A) et revenir en 1.
- 3 - si c'est OUI faire la mise à jour (ADDITIONNER).
- 4 - passer au mouvement suivant (ENTRER M) et revenir en 1.

Au début faire "ENTRER A" et "ENTRER M", qui sont les conditions préalables au test "A = M ?".

On comprend alors que l'apprentissage du processus de commande en phase 3 soit difficile et puisse entraîner le retour d'erreurs caractéristiques des

phases précédentes : il faut procéder à un redécoupage de l'action, à une redéfinition des buts et sous buts.

Comme le souligne Hoc (1980), "l'élaboration des règles des processus s'est avérée correspondre à un mécanisme d'adaptation de procédures disponibles au préalable. Ainsi les processus n'ont pas été élaborés à partir de règles élémentaires du fonctionnement du dispositif : ces dernières ont été acquises en référence à une procédure connue". Il est remarquable en effet que l'élaboration de la procédure soit longue et difficile alors que l'effet des commandes est simple, que s'il n'a pas été compris à partir des instructions, il a dû l'être grâce à la pratique acquise dans la première phase. Or, en seconde phase, la construction de la procédure s'avère difficile alors que les mêmes commandes sont utilisées et il en va de même en troisième phase où le test "A = M ?" n'avait certes pas été utilisé en tant que tel mais correspond à une opération que le sujet réalisait déjà dans la seconde phase où il avait à comparer les contenus des cases A et M.

La procédure n'est donc pas déduite de la connaissance des règles de fonctionnement, elle est le fruit de modifications apportées à une procédure connue pour la rendre compatible avec les contraintes du dispositif. Ces modifications tendent à être minimales : ainsi en phase 2 il était possible d'élaborer une procédure centrée sur le traitement des mouvements comme en phase 1, néanmoins les sujets semblent plutôt adopter une procédure centrée sur le traitement des pièces, qui est plus proche de la procédure utilisée en phase 1. Les sujets font les remises en cause minimales compatibles avec les nouvelles contraintes mais ces remises en cause peuvent concerner l'organisation générale de la procédure (la définition des buts et des sous-buts), comme en phase 3, les modalités de réalisation de sous-buts comme en phase 1 ou les deux comme en phase 2.

Si la procédure n'est pas déduite des règles de fonctionnement alors que celles-ci semblent connues, on peut faire l'hypothèse que la connaissance des règles de fonctionnement n'est pas suffisante pour élaborer la procédure et qu'il manque un élément essentiel lié à la redéfinition de nouveaux sous-buts compatibles avec les contraintes du dispositif. C'est en effet, la réorganisation de la structure générale de la procédure imposée par le non-disponibilité de certaines informations plus que l'utilisation des instructions qui fait la difficulté de la situation étudiée par Hoc.

4 - Une expérience de formation basée sur la logique de l'utilisation

Bisseret et Enard (Enard, Bisseret 1969, Bisseret, Enard 1969, Enard 1972) ont mis au point, en vue de la formation des contrôleurs aériens, une méthode de formation faisant appel à l'enseignement programmé et dans laquelle la structuration des unités programmées suit la logique de l'utilisation. C'est là un point de vue fort différent de celui habituellement adopté en enseignement programmé, dans la mesure où ce dernier repose sur une analyse de la matière centrée sur la logique du contenu et de l'exposition : le nouveau est construit à partir du connu, les notions contenues dans la définition doivent avoir été présentées auparavant, ceci correspond à une organisation de la connaissance en forme d'arbre exprimant les relations de dépendance qui existent entre les notions dans le cadre du mode d'exposition choisi et ne prenant en compte que les relations internes, relatives aux éléments de connaissance, à l'exclusion des relations externes faisant intervenir les contextes d'utilisation.

Bisseret et Enard (1969, 1972) avaient dans un premier temps construit un enseignement basé sur ces principes et utilisant des tâches-contrôle classiques destinées à tester l'acquisition des éléments de connaissance : par exemple faire dessiner ou compléter une carte vierge pour vérifier l'apprentissage de la carte aérienne. Il est apparu qu'à l'examen les apprentis savaient fort bien reproduire la carte aérienne mais commettaient lors du travail des erreurs qui révélaient aux yeux des instructeurs une méconnaissance de la carte. En fait, ils avaient bien une connaissance de la carte mais une connaissance inadéquate au contexte du travail.

Bisseret et Enard (1969) soulignent la différence entre la logique de l'utilisation et une logique de l'exposition centrée sur le contenu de ce qui est à enseigner : "Nous distinguerons pour chaque unité élémentaire composante du travail une structure spécifique indépendante du but du système de travail (que nous nommerons logique de construction) et une structure conséquente de ce but (que nous nommerons logique d'utilisation)". La logique de construction (comme la logique du fonctionnement dans le cas d'un appareil) est une présentation de la connaissance qui contient (en ce sens qu'elle permet de les dériver) toutes les informations nécessaires à l'ensemble des utilisations potentielles.

Ainsi, le code des aérodromes obéit à une logique définie : chacune des lettres qui le composent donne certaines informations permettant d'identifier le secteur international, l'Etat, la localisation géographique.

Ces informations servent à différents types d'utilisations, par exemple décoder le nom de l'aérodrome sur la fiche de progression de l'avion mais aussi classer l'avion du point de vue des interventions à prévoir : il fera un simple survol, il demandera à descendre. Ces indications associées à d'autres (niveau de l'avion, liste des balises) permettent de prévoir la trajectoire de l'avion, d'anticiper d'éventuels conflits avec la trajectoire d'autres avions.

Dans la logique de construction, le code des aérodromes n'a aucune relation directe avec le niveau de l'avion ou la liste des balises : il en a dans la logique de l'utilisation. C'est pourquoi dans l'élaboration du programme d'enseignement Bissere et Enard présentent chaque unité dans les différents contextes d'utilisation, dans le contexte des différentes tâches dans lesquelles elle intervient et en association avec les autres unités de connaissance en jeu.

Ainsi, la notion de zone réglementée apparaît dans un premier contexte, qui correspond à la logique de construction : il s'agit d'apprendre les caractéristiques de chaque zone (niveau plancher et niveau plafond autorisés, personnes à aviser) et les questions posées concernent la mémorisation de ces informations. Cette notion intervient ensuite dans le contexte de la tâche de coordination entre contrôleurs, elle interviendra encore dans des situations de demande de changement de niveau de la part de pilotes, ce qui suppose que l'opérateur vérifie si c'est compatible avec les contraintes de la zone, qu'il se préoccupe de la coordination en avisant les personnes intéressées.

Le principe de cet enseignement consiste à passer un minimum de temps à une présentation centrée sur la matière à enseigner et à en ménager l'essentiel à l'apprentissage des différentes utilisations qui peuvent être faites des diverses informations dans les tâches qui constituent le travail de l'opérateur.

Ce parti pris repose sur l'idée qu'il ne suffit pas de développer en mémoire la forme d'organisation de la connaissance qui est privilégiée, en ce sens qu'elle permet en principe de déduire toutes les autres, mais qu'il faut

faire apprendre à l'élève ou à l'apprenti les diverses utilisations qu'il peut faire de cette connaissance (au lieu de supposer qu'il le découvrira tout seul) et ainsi de construire en mémoire d'autres formes d'organisation dans lesquelles les connaissances sont liées à des tâches et à des procédures et ne sont liées entre elles qu'à travers les procédures dans lesquelles elles interviennent.

Le fait qu'un tel enseignement se soit révélé efficace non seulement quant à la réduction du temps moyen d'obtention de la qualification mais aussi quant à la réduction des différences interindividuelles dans le temps d'apprentissage, donne assurément du crédit à cette idée.

5 - Les difficultés cognitives liées au passage de la logique du fonctionnement à la logique de l'utilisation.

Les expériences de Friemel et al (1982a et b) Hoc (1980) Bisseret et Enard (1969, 1972), laissent penser que le passage de la connaissance du fonctionnement à la découverte des règles d'utilisation constitue une difficulté importante pour les sujets bien qu'a priori toutes les connaissances impliquées dans l'utilisation soient contenues dans la connaissance des règles de fonctionnement il convient donc de se demander à quoi tiennent ces difficultés.

5.1 L'activité déductive impliquée dans le passage des règles de fonctionnement aux règles d'utilisation et les difficultés liées à cette déduction

On ne remarque pas suffisamment que passer des connaissances concernant le fonctionnement, d'un dispositif à leur utilisation dans un contexte défini pour réaliser un objectif particulier implique un travail déductif important et difficile du point de vue cognitif.

Les règles de fonctionnement sont du type suivant :

Si on provoque l'exécution d'une commande C1 on obtient les effets E1, E2.

Par contre, les règles d'utilisation ont la forme suivante :

Si on veut avoir E1, on peut faire C1.

Autrement dit, les règles de fonctionnement sont du type :

Si "p" alors "q".

Et les règles d'utilisation

Si l'objectif est "q" alors on peut faire "p".

En d'autres termes, il s'agit de passer du conséquent à l'antécédent. Or, le passage du conséquent à l'antécédent n'est simple que dans un seul cas, celui de l'équivalence, celui où on a à la fois "p implique q" et "q implique p". Alors on peut permuter simplement conséquent et antécédent. Cela n'est évidemment pas légitime lorsqu'on a une implication : de "p implique q" on ne peut pas tirer : si on a q alors on doit avoir p, mais seulement : on peut avoir p.

Dans les situations de raisonnement syllogistique l'implication est souvent assimilée à l'équivalence : les sujets considèrent qu'une situation où non-p et q sont vrais infirme l'implication. Dans le cas où l'implication exprime une règle de fonctionnement ("p" est une commande, "q" un effet) l'assimilation de l'implication à l'équivalence s'exprime par la déduction suivante : pour avoir q il faut faire p, au lieu de : pour avoir "q" on peut faire "p".

Il est possible qu'une assimilation de ce type soit responsable du phénomène appelé fixité fonctionnelle, très connu dans les recherches classiques sur la résolution de problèmes. Soit, par exemple, l'énoncé : si on relie les bornes d'un générateur par un fil métallique, alors on établit un circuit électrique. On est tenté d'en conclure que pour établir un circuit il faut nécessairement un fil métallique et cela pourrait expliquer que des sujets à qui on demande de compléter un circuit et qui ne disposent que d'un tournevis, ont des difficultés à résoudre le problème (Danks et Gluksberg 1966). L'assimilation de l'implication à l'équivalence revient à associer chaque utilisation à un seul instrument, comme dans le raisonnement expérimental elle revient à associer chaque effet à une seule cause.

Le cas où la règle de fonctionnement s'exprime par une implication du type "p implique q" est le plus simple. Le plus souvent l'effet d'une commande est multiple et on a un énoncé du type : "p implique q et r".

Dans ce cas le travail de déduction est plus complexe : si l'on a comme objectif q , alors on peut l'atteindre au moyen de p mais il faut remarquer qu'on aura nécessairement r en même temps que p et donc vérifier si r n'aura pas d'effet néfaste pour la suite de l'action. Si oui, il faudra préalablement prendre ses précautions pour que l'effet r ne soit plus gênant au moment où la commande p sera exécutée ou alors chercher une autre commande permettant d'avoir l'effet q .

Prenons un exemple : dans la calculette Texas TI 30, comme dans beaucoup de modèles ayant une mémoire, deux instructions peuvent être utilisées pour stocker un résultat en mémoire. Ce sont les commandes STO et EXC dont le fonctionnement peut être décrit par les règles suivantes :

Si on appuie sur la touche STO alors :

- 1 - la donnée présente à l'affichage est inscrite en mémoire.
- 2 - la donnée qui était précédemment en mémoire est détruite.

Si on appuie sur la touche EXC alors :

- 1 - la donnée présente à l'affichage est inscrite en mémoire.
- 2 - la donnée précédemment en mémoire est inscrite à l'affichage.

De ces règles de fonctionnement on peut inférer les règles d'utilisation suivantes :

Si on veut conserver un résultat en mémoire et si on n'a plus besoin de ce qui est en mémoire, alors on peut utiliser la touche STO.

Si on veut conserver un résultat en mémoire et si on a encore besoin de ce qui est en mémoire, alors il faut utiliser la touche EXC.

Sans être très complexe, la déduction de ces règles d'utilisation peut présenter des difficultés : cela n'est pas étonnant, dans la mesure où il y a largement autant d'informations à traiter et de transformations à effectuer que dans un syllogisme.

Ce travail déductif passe inaperçu car il n'est jamais explicité, et s'il devient automatisé chez l'opérateur expérimenté, ce n'est pas le cas chez le novice.

Les exemples que nous avons présentés restent simples dans la mesure où la règle d'utilisation est déduite d'un petit nombre de règles de fonctionnement (une ou deux), mais il peut arriver, et ce n'est pas rare, que des possibilités d'utilisation très intéressantes mettent en jeu des connaissances qui sont d'une part nombreuses et souvent très dispersées dans un exposé fait du point de vue de la logique du fonctionnement. Je voudrais citer l'exemple d'une possibilité d'utilisation de la calculette Texas TI 30 dont nous avons pu constater (Friemel Weil, Richard 1982 a et b) qu'elle n'a été découverte spontanément par aucun des sujets de l'expérience (plus d'une centaine). Dans ce type de machine on peut faire deux calculs simultanément, par exemple, pour reprendre le problème décrit page 18, faire à la fois la somme d'une suite de valeurs et la somme des carrés. On peut faire la somme des données en utilisant les possibilités d'addition en mémoire, grâce à l'instruction SUM (addition algébrique en mémoire), et la somme des carrés sur le registre d'affichage en utilisant les opérateurs arithmétiques. Cette possibilité d'utilisation n'a été découverte, et encore par moins de la moitié des sujets, que lorsqu'on leur a donné la contrainte d'introduire une seule fois les données en machine.

Cette possibilité d'utilisation de la machine d'une part est contraire aux habitudes du calcul à la main mais d'autre part implique pour être déduite, que l'on fasse appel à un nombre important de connaissances concernant le fonctionnement. Ce sont :

- 1 - l'effet de la commande SUM
- 2 - l'effet des commandes arithmétiques + , x , élévation au carré, mais, aussi et surtout :
- 3 - le fait qu'une opération arithmétique telle que l'addition n'est pas exécutée immédiatement mais ne l'est que lors de la commande ultérieure d'un autre opérateur + (ou -) ou de la touche =.
- 4 - le fait que lorsqu'une donnée présente à l'affichage est ajoutée en mémoire, elle reste présente à l'affichage (et donc on peut opérer sur elle).

Toutes ces connaissances sont nécessaires pour déduire cette possibilité d'utilisation qui pratiquement se présente de la façon suivante :

- introduire la première donnée.
 - faire $\boxed{\text{SUM}}$ ce qui ajoute en mémoire cette donnée (connaissance 1).
 - faire $\boxed{x^2}$ pour élever au carré cette donnée (connaissance 2), ce qui est permis du fait que la donnée est présente à l'affichage (connaissance 4).
 - faire $\boxed{+}$ pour préparer l'addition du carré de la première donnée à celui de la seconde (connaissance 2).
 - introduire la deuxième donnée (connaissance 2).
 - faire $\boxed{\text{SUM}}$ pour additionner cette donnée en mémoire. Ceci est possible car la machine n'exécute pas l'addition immédiatement après l'introduction du second opérande (connaissance 1 et connaissance 3).
- etc.

Les connaissances sont données de façon très dispersées dans un exposé sur le fonctionnement : les deux premières concernent les opérateurs, le troisième les règles de priorité et la quatrième n'est pas donnée de façon explicite et n'a d'ailleurs pas à l'être dans un exposé sur le fonctionnement. Dans un tel exposé l'effet des commandes est décrit comme un changement d'état et il n'y a lieu d'indiquer que ce qui change, étant entendu que tout le reste est constant. Il suffit de dire qu'à la suite de la commande $\boxed{\text{SUM}}$ la valeur affichée est ajoutée en mémoire ; si l'on ne parle pas du registre d'affichage, c'est que rien n'est changé et donc que la valeur précédente reste affichée. Ce n'est pourtant pas ce que comprennent un certain nombre de sujets : ils considèrent qu'elle a disparu, ce qui serait d'ailleurs le cas s'il s'agissait d'une addition réalisée au moyen de la commande $\boxed{+}$.

Dans un exposé cohérent sur le fonctionnement les connaissances nécessaires à la déduction des utilisations possibles sont en général dispersées et parfois implicites, parce que la logique de l'exposition a ses lois propres. Pour déduire des règles d'utilisation il ne suffit pas d'avoir compris les règles de fonctionnement, il faut opérer un travail de déduction important, qui

suppose non seulement des capacités de déduction suffisantes mais la possibilité de maintenir en mémoire de travail les diverses connaissances nécessaires à la dérivation.

5.2 La nécessité pour le sujet de modifier sa représentation de la tâche

Réaliser à l'aide d'un dispositif une tâche que l'on sait résoudre par ailleurs, constitue une situation problème typique. On sait réaliser sans difficulté la tâche impliquée dans la "Tour de Hanoï" si l'on n'est pas limité par le nombre d'emplacements et par l'impossibilité de déplacer plus d'un disque à la fois. C'est aussi le cas pour le casse-tête des "missionnaires et des cannibales" si l'on peut avoir un bateau suffisamment grand. Les problèmes classiques, ce n'est pas une coïncidence, sont des situations où la procédure habituelle pour réaliser la tâche est rendue impossible par les contraintes qui sont imposées. Dans le cas d'un dispositif les contraintes sont représentées par les commandes, qui définissent les actions élémentaires possibles. Comme dans le cas des problèmes classiques, il faut modifier sa représentation du problème, définir de nouveaux buts et sous-buts.

Dans un problème tel que celui de la "Tour de Hanoï" ou celui des missionnaires et cannibales", le sujet a une idée assez claire de l'ensemble des actions possibles. Ce n'est pas le cas pour un dispositif apparemment simple tel qu'une calculatrice. Comme on vient de le voir, certaines possibilités d'utilisation sont très difficiles à découvrir. Il n'est pas étonnant dans ces conditions que les sujets aient de grandes difficultés à définir des sous-buts appropriés : l'ensemble des actions possibles est pour eux très mal défini : certaines sont exclues par les sujets alors qu'elles sont permises, d'autres ne sont même pas envisagées, certaines enfin sont considérées comme possibles alors qu'en fait elles sont interdites par le dispositif.

La difficulté essentielle dans l'utilisation d'un dispositif d'aide au traitement de l'information nous paraît résider en ceci que l'opérateur n'étant pas à même de se construire à partir de sa connaissance du fonctionnement, même si celle-ci est bonne, une connaissance suffisante des possibilités réelles de la machine, de ce qu'elle permet de faire et de ce qu'elle ne permet pas, il n'arrive pas à définir des sous-buts appropriés, à construire une organisation de la tâche qui soit compatible avec les contraintes du dispositif.

6 - Vers l'apprentissage de règles d'utilisation

S'il est difficile de déduire de la connaissance du fonctionnement tout ce que la machine permet de faire, il devrait être possible de faciliter l'apprentissage en formulant des règles d'utilisation qui précisent les actions qu'il est possible de réaliser à l'aide de la machine et la façon de le faire.

La difficulté est de définir quelles doivent être ces actions. Elles doivent être susceptibles de constituer des sous-buts, d'être envisagées par le sujet comme des moyens de réaliser la tâche. Les transformations qui définissent l'effet de chaque commande dans le cadre d'un modèle de fonctionnement du dispositif ne peuvent en elles-mêmes constituer des buts pour le sujet : elles sont définies par rapport à la sémantique du dispositif (ce sont des changements d'état du dispositif). Il faut définir des actions qui aient un sens pour le sujet, un sens dans le langage des actions qu'il prend comme objectifs dans les conditions habituelles de réalisation de la tâche.

Certes, on peut à la limite définir complètement les procédures à mettre en oeuvre pour les diverses tâches réalisables à l'aide du dispositif. Cela est vrai pour beaucoup d'appareils d'usage quotidien (appareils radio, HI-FI, appareils ménagers, appareils pour le bricolage) pour lesquels l'ensemble des tâches est limité. Cela est vrai, sans doute aussi, pour les tâches complexes pour lesquelles on connaît assez bien les procédures utilisées par les opérateurs : ainsi la bonne connaissance des règles de décision des contrôleurs aériens a permis à Bissieret et Enard d'élaborer un programme de formation efficace basé sur l'utilisation des connaissances plus que sur leur organisation interne.

Dans le cas général on ne connaît pas à l'avance l'ensemble des tâches auxquelles peut être utilisé un dispositif et il n'est donc pas question d'élaborer un programme d'exécution pour chacune. On peut donner des exemples, comme on le fait ordinairement dans les manuels, mais cela ne suffit pas.

Il faut définir des actions qui ne correspondent pas à des tâches mais qui constituent des composantes à partir desquelles puissent se définir les tâches et qui par ailleurs, comme on l'a dit, puissent constituer des sous-buts réalistes pour le sujet, et donc soient des actions dont le résultat ait un sens dans l'analyse que le sujet fait de la tâche.

Nous ne sommes pas en mesure de caractériser plus précisément en termes généraux ce que doivent être ces actions. Nous présentons ci-dessous, à titre d'exemple, des règles d'utilisation concernant une calculatrice du type algébrique que nous avons étudiée (Friemel, Richard, Weil Barais 1982), et dont nous avons présenté le modèle de fonctionnement au paragraphe 3 (p. 17).

Dans cet exposé, il nous semble intéressant de distinguer différentes possibilités d'utilisation, par exemple :

- on peut faire des opérations en utilisant les opérateurs arithmétiques figurant sur les touches du clavier $\boxed{+}$ $\boxed{-}$ $\boxed{\times}$ $\boxed{\div}$ $\boxed{\frac{1}{x}}$? etc. Le résultat de l'opération apparaît sur le registre d'affichage.
- on peut faire certaines opérations arithmétiques (addition et soustraction) en utilisant la mémoire : le résultat de l'opération est alors dans la mémoire.
- on peut utiliser la mémoire pour stocker le résultat d'une opération et le rappeler pour servir d'opérande dans une opération ultérieure et cela autant de fois qu'on le veut.
- on peut faire simultanément deux opérations sur une même donnée : l'une en utilisant les opérateurs arithmétiques et le registre d'affichage, l'autre en utilisant la mémoire.

Nous présentons ci-dessous une formulation possible des règles d'utilisation correspondant à ces différents objectifs. Il en existe sans doute de meilleures dans la perspective d'un apprentissage. La formulation présentée n'a pas été testée elle est donnée à titre d'illustration pour montrer ce que peuvent être des règles d'utilisation et en quoi elles diffèrent des règles de fonctionnement.

6.1 Stocker un résultat en mémoire et l'utiliser ultérieurement.

- Pour stocker un résultat en mémoire :

Si l'on n'a pas besoin de ce qui est en mémoire, on utilise la commande STO le résultat affiché est copié en mémoire et ce qui était précédemment en mémoire disparaît.

Si l'on a besoin de ce qui est en mémoire, on utilise la commande EXC : le résultat affiché est en mémoire, le résultat qui était en mémoire est affiché et on peut opérer sur lui avec les opérateurs arithmétiques.

- Pour rappeler un résultat de la mémoire :

Si l'on n'a pas besoin du résultat affiché, on utilise la commande RCL : le résultat stocké en mémoire est copié dans le registre d'affichage mais il reste disponible en mémoire pour des rappels ultérieurs.

Si l'on a encore besoin du résultat affiché, on utilise la commande EXC : le résultat affiché est copié en mémoire et inversement le résultat qui était en mémoire est copié dans le registre d'affichage. On peut ainsi permuter les opérandes d'une opérations (voir p. 40).

On remarquera que la première partie de l'énoncé donne la règle d'utilisation, la seconde partie, celle qui suit le signe de ponctuation ":", donne la règle de fonctionnement qui justifie la règle de l'utilisation.

6.2 Faire des opérations en mémoire

- Pour faire une addition en mémoire :

Il faut que le premier opérande soit en mémoire : s'il n'y est pas, le mettre en l'introduisant au clavier et en utilisant la commande STO.

Introduire le second opérande et appuyer sur la commande SUM : le second opérande est ajouté au premier et la somme est en mémoire.

- Pour faire une soustraction en mémoire :

Il faut que le premier opérande soit en mémoire (cf. supra).

Introduire le second, faire la commande +/- puis la commande SUM: le nombre introduit est affecté du signe -, puis est soustrait du nombre en mémoire ; la différence est en mémoire.

6.3 Faire des opérations sur le registre d'affichage

Les règles d'utilisation sont plus complexes dans ce cas, car les règles de fonctionnement le sont également, et elles dépendent des différents cas d'utilisation. On peut énoncer les règles générales suivantes :

- R1 : faire en sorte que le premier opérande (l'opérande, s'il s'agit d'une opération à un seul opérande) soit présent dans le registre d'affichage. Il peut l'être de 3 façons : il est introduit au clavier, il est rappelé de la mémoire, il est le résultat d'un calcul précédent.
- R2 : Introduire l'opérateur une fois que le premier opérande est dans le registre d'affichage et juste avant d'introduire ou de calculer le second opérande.
- R3 : Pour les opérations à 2 opérandes, faire en sorte que le second opérande soit présent dans le registre d'affichage, c'est-à-dire : soit l'introduire au clavier, soit le rappeler de la mémoire, soit faire en sorte que le calcul dont le résultat constitue le second opérande soit effectué au moment où sera exécutée l'opération.
- R4 : Faire exécuter l'opération, (pour les opérations à 2 opérandes seulement car pour les opérations à un opérande, l'exécution suit automatiquement l'introduction de l'opérateur).

Pour énoncer des règles plus spécifiques qui restent relativement simples, il semble commode de distinguer différents cas d'utilisation. Nous avons choisi les suivants :

- . on veut faire une opération isolée.
- . on veut faire une suite d'opérations enchaînées (sans sortir les résultats intermédiaires) et on ne souhaite pas utiliser les possibilités offertes par les règles de priorité d'exécution des opérations (on ne les connaît pas ou on les connaît mal).
- . on veut faire une suite d'opérations enchaînées, on veut utiliser les possibilités offertes par les règles de priorité d'exécution des opérations mais non celles offertes par les parenthèses.
- . On veut faire une suite d'opérations enchaînées et on souhaite utiliser toutes les possibilités offertes par la machine.

La justification de cette distinction tient à ce que les connaissances impliquées pour la compréhension du fonctionnement ne sont pas du même niveau dans ces différents cas : celles concernant les règles de priorité et les parenthèses sont loin d'être élémentaires et sont mal maîtrisées par beaucoup d'élèves au terme de l'enseignement secondaire.

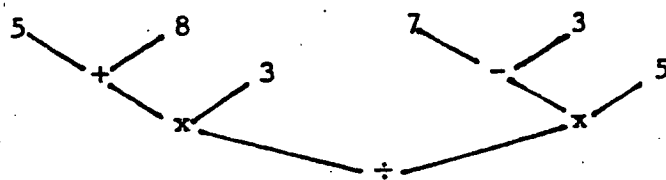
6.3.1 Faire une opération isolée

Dans ce cas la règle est simple :

- s'il s'agit d'une opération à un opérande :
introduire l'opérande, introduire l'opérateur.
- s'il s'agit d'une opération à deux opérandes :
introduire le premier opérande, introduire l'opérateur, introduire le second opérande, faire exécuter l'opération à l'aide de la commande =

6.3.2 Faire une chaîne d'opérations sans utiliser les règles de priorité ni les parenthèses

On peut représenter la suite des opérations à effectuer par un graphe tel que celui dessiné ci-dessous qui signifie : ajouter 8 à 5 multiplier la somme par 3 puis diviser le résultat par le produit de la différence (7 - 3) par 5.



Les opérations sont effectuées de haut en bas et de gauche à droite.

Dans ce cas il faut utiliser les règles suivantes :

- si le premier opérande est une donnée, l'introduire.
- si le premier opérande est en mémoire, le rappeler avec la commande RCL
- si le premier opérande est le résultat d'un calcul précédent, il se trouve affiché et il n'y a rien à faire pour l'introduire dans le registre d'affichage.

- si le second opérande est une donnée, introduire l'opérateur, introduire la donnée, faire [=] .
- si le second opérande est en mémoire introduire l'opérateur et rappeler le second opérande (avec RCL ou EXC) et faire [=] .
- si le second opérande est le résultat d'un calcul (c'est le cas de la division en bas du graphe) celui-ci doit être préalablement exécuté.
 - . transférer en mémoire le premier opérande avec la commande STO.
 - . faire exécuter le calcul qui donne le second opérande.
 - . permuter les opérandes au moyen de la commande EXC pour amener le premier opérande dans le registre d'affichage (cette opération n'est pas nécessaire dans le cas d'opérations comme l'addition ou multiplication, dans lesquelles l'ordre des opérandes est indifférent).
 - . introduire l'opérateur, rappeler l'autre opérande de la mémoire (RCL) et faire [=] .

Pour exécuter les calculs du graphe ci-dessus on fera :

[5] [+] [8] [=] [x] [3] [=] [STO] [7] [-] [3] [=] [x] [5] [=] [EXC] [÷] [RCL] [=]

Les trois premières règles constituent des spécifications de la règle R1, les trois dernières des spécifications des règles R2, R3 et R4. Les règles générales expriment des sous buts, les règles spécifiques indiquent comment réaliser ces sous buts selon les différents contextes (nature des opérandes).

6.3.3 Faire une chaîne d'opérations en utilisant les règles de priorité algébriques mais non les parenthèses.

La formulation habituelle des règles de priorité algébriques est directement utilisable quant il s'agit d'interpréter une formule, en particulier de déduire l'ordre d'exécution des opérations à partir de la formule. Ces règles peuvent être présentées ainsi :

- Les opérations ont une hiérarchie dans l'ordre d'exécution :
 - priorité 1 (la plus élevée) : y puissance x.
 - priorité 2 : multiplication, division.
 - priorité 3 : addition, soustraction.

- De deux opérations qui n'ont pas le même niveau de priorité, celle qui a le niveau de priorité le plus élevé est exécutée la première.
- De deux opérations qui ont le même niveau de priorité, c'est celle qui précède dans l'ordre d'écriture qui est exécutée la première.

Si l'on connaît l'ordre de calcul et s'il s'agit de passer du graphe qui le représente à la formule, ce qui correspond à l'utilisation qui est faite quand on se sert d'une machine à calculer, on peut dériver les règles d'utilisation suivantes, qui s'ajoutent aux règles décrites au paragraphe 6.3.2.

- Si l'opération qui suit est de priorité supérieure, faire [=] après avoir introduit le second opérande. L'opération en cours doit, en effet être exécutée avant l'opération qui suit, puisque son résultat sert de second opérande à cette dernière.
- Si l'opération qui suit est de priorité égale ou inférieure il n'y a pas besoin de faire [=] car de toute façon elle sera exécutée avant l'opération qui suit.

Ainsi l'utilisation de ces règles permet seulement de faire l'économie de la commande [=] dans certains cas.

6.3.4 Faire une chaîne d'opérations en utilisant les règles de priorité et les parenthèses

Les règles concernant les parenthèses sont également exprimées en général sous une forme directement utilisable quand il s'agit d'interpréter une formule. On peut les énoncer ainsi :

- une expression entre parenthèses a la priorité d'exécution la plus élevée : les opérations qui sont à l'intérieur de la parenthèse sont exécutées avant l'opération qui correspond à l'opérateur qui précède (et évidemment celle qui correspond à l'opérateur qui suit).
- s'il y a plusieurs parenthèses emboîtées, ce sont les opérations qui sont dans la parenthèse la plus intérieure qui sont exécutées les premières.
- à l'intérieur d'une parenthèse les opérations sont exécutées selon les règles de priorité indiquées précédemment.

Quand on veut faire la démarche inverse, passer d'un schéma de calcul à une séquence de commandes, les règles d'utilisation sont en fait assez complexes : plusieurs cas d'utilisation doivent être distingués car ils correspondent à des objectifs différents.

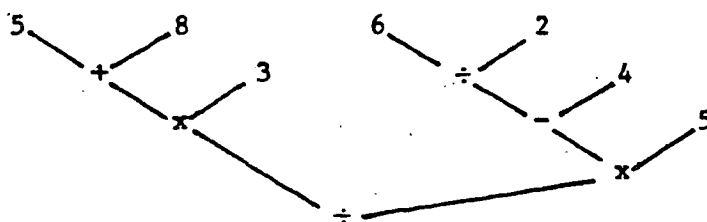
D'une façon générale les parenthèses permettent de modifier l'ordre d'exécution qui résulte des règles de priorité et ne sont utiles que lorsque l'opérande est lui-même le résultat d'un calcul.

Si c'est le premier opérande qui est le résultat d'un calcul, on veut que le calcul soit exécuté avant que l'on introduise l'opérateur de l'opération en cours. Si cette dernière est de priorité plus élevée que la précédente, on peut mettre le premier opérande entre parenthèses mais on peut aussi utiliser la commande $\boxed{=}$ qui dans ce contexte a le même effet. Cette solution est plus simple du point de vue des règles d'utilisation, puisqu'il suffit d'appliquer les règles énoncées au paragraphe précédent (6.3.3.).

Quand c'est le second opérande de l'opération en cours qui est le résultat d'un calcul, on veut également que ce calcul soit exécuté avant l'opération en cours, mais comme ce calcul suit l'introduction de l'opérateur il faut mettre le calcul entre parenthèses, s'il y a dans le calcul une opération de priorité égale ou inférieure, car dans ce cas celle-ci serait exécutée après l'opération en cours.

A l'intérieur du calcul par contre, comme dans le cas où le résultat du calcul constitue le premier opérande, on veut que les opérations soient exécutées dans l'ordre où elles sont introduites : si une opération est suivie par une opération de priorité supérieure, la première doit être mise entre parenthèses (une parenthèse doit être ouverte avant le premier opérande et fermée après le second).

Prenons comme exemple le graphe de calcul suivant :



La branche de gauche, qui constitue le premier opérande de la division du bas du graphe, ne présente pas de difficulté. Le second opérande de cette division est le résultat d'un calcul ; ce calcul doit être mis entre parenthèses puisqu'il y a dans le calcul des opérations de priorité égale ou inférieure. A l'intérieur du calcul la division $6 : 2$ n'a pas à être mise entre parenthèses car l'opération qui suit $(-)$ est de priorité inférieure. Par contre, cette soustraction doit être mise entre parenthèses car l'opérations suivante, une multiplication a une priorité plus élevée : il faut ouvrir une parenthèse avant le premier opérande de la soustraction c'est-à-dire avant 6 et la fermer après le second (après 4). Pour l'ensemble du calcul la suite des commandes est :

5 + 8 = x 3 ÷ ((6 ÷ 2 - 4) x 5) =

On voit que ces règles sont assez complexes et qu'en fait tout le bénéfice de leur application tient à une économie dans les parenthèses. Si l'on n'a pas ce souci de l'économie, qui ne se justifie vraiment que dans la programmation on peut se contenter d'une règle très simple: quant un opérande est le résultat d'un calcul, on met l'opérande entre parenthèses et à l'intérieur du calcul on met toutes les opérations entre parenthèses.

Les exemples précédents permettent de faire apparaître les caractéristiques des règles d'utilisation.

- Ces règles sont dérivées des règles de fonctionnement mais elles font intervenir explicitement des objectifs d'action. Nous avons supposé que l'utilisateur se proposait de passer d'un schéma de calcul connu à son écriture dans le langage de commande de la machine et que le calcul est composé d'opérations que l'utilisateur sait analyser en termes d'opérateurs et d'opérandes.
- L'organisation des règles d'utilisation dépend autant des objectifs d'action que des connaissances concernant le fonctionnement. Nous avons vu ainsi que que la réalisation de l'objectif d'action le plus complexe faire des calculs en chaîne, pouvait être rempli sans faire appel aux règles de fonctionnement concernant la priorité des opérations et en se contentant d'une règle minimale concernant les parenthèses. Le bénéfice

apporté par la prise en compte des règles complexes de fonctionnement tient pour l'essentiel à une économie de commandes, qui dans certaines utilisations doit être considérée, mais dans beaucoup d'autres importe peu. On peut donc différencier les règles d'utilisation en distinguant d'une part les connaissances du fonctionnement sur lesquelles elles reposent et d'autre part les types d'objectifs auxquels elles peuvent répondre.

- Les règles d'utilisation sont utilisables directement pour élaborer une procédure. Si le système de règles est complet, on peut, étant donné un schéma de calcul, dériver une procédure d'exécution en appliquant directement les règles précédentes. On peut évidemment pour ce faire partir des règles de fonctionnement, mais comme elles sont de forme "si on fait ceci dans tel contexte, il se produit cela", on sait seulement ce qui se passe quand on fait l'action projetée. Il faut en outre se demander si ce qui se passe correspond à ce que l'on cherche à faire et, sinon, essayer autre chose. Par exemple, je veux additionner 5 et 8 et multiplier le résultat par 3 : en appliquant les règles de fonctionnement, je peux savoir que si je fais $\boxed{5} \boxed{+} \boxed{8} \boxed{\times} \boxed{3}$, la multiplication sera exécutée avant l'addition, ce qui ne correspond pas à ce que je cherche. Je dois donc envisager une autre façon de faire : les règles de fonctionnement permettront de tester si elle convient mais non de la trouver. Il faut pour cela utiliser une heuristique générale de raisonnement du type : si je sais que $p \supset q$ et que je souhaite q , alors faire p . Je sais que : si une expression est entre parenthèses (p) alors elle exécutée avant l'opération qui suit (q). Je fais en sorte que p soit vrai : c'est-à-dire que je mets l'expression entre parenthèse.
- Les règles d'utilisation sont des règles qui peuvent être abstraites à partir de procédures mises au point par d'autres moyens, par exemple en faisant des essais, en les testant et en les corrigeant. On peut penser que le savoir de l'expert consiste en partie à forger des règles de ce type appropriées à la réalisation des tâches dont il a l'expérience.

7 - Remarques finales

L'exposé qui précède a pour objectif principal de soulever des questions. Nous les résumons ici :

- 1 - Il paraît important de distinguer deux modes d'exposition : l'un est centré sur le fonctionnement du dispositif : il repose sur une description de celui-ci sous forme d'automate, et il décrit l'effet de chaque commande en termes de changements d'état. L'autre est centré sur des résultats correspondant à des objectifs d'action qui aient un sens en dehors du dispositif et qui puissent être envisagés comme sous-buts dans une tâche complexe. Une partie importante des difficultés rencontrées dans la compréhension et l'utilisation des notices tient vraisemblablement, c'est notre hypothèse, au fait que ces deux points de vue sont mal distingués : on trouve dans celles-ci d'une part des explications qui tiennent au fonctionnement mais qui sont insuffisantes pour fournir un modèle de fonctionnement en termes d'automate et d'autre part des exemples d'utilisation extrêmement spécifiques, qui sont supposés faciliter la compréhension mais qui en réalité se réfèrent à un point de vue complètement différent de celui de la logique du fonctionnement.
- 2 - Déduire des règles d'utilisation de la connaissance des règles de fonctionnement n'est pas une chose simple lorsqu'une commande a des effets multiples et différents selon le contexte. La difficulté du raisonnement impliqué dans une telle déduction est probablement sous-estimée : l'obtention d'un résultat correspondant à un objectif d'action fait appel le plus souvent à plusieurs règles de fonctionnement et dans ce cas il faut que ces différentes règles soient présentes en mémoire en même temps et que d'autre part les déductions appropriées soient réalisées. En d'autres termes, ce qui est difficile pour le novice c'est probablement de savoir ce que le dispositif permet de réaliser en fait d'actions qui soient à la fois réalisables à l'aide des commandes disponibles et envisageables comme des objectifs possibles, comme sous-buts qu'il puisse se donner dans une tâche complexe. Même s'il connaît l'effet des commandes, cette connaissance n'est pas utilisable immédiatement car en général l'effet d'une commande ne correspond pas au résultat d'une action qui soit équivalente à un sous-but dans la programmation habituelle de l'action.

3 - Le problème de l'apprentissage doit être pensé en relation avec le type d'utilisation qui doit être fait. On peut distinguer deux grandes formes d'utilisation :

- la définition d'une suite de commandes permettant d'obtenir un résultat souhaité.
- le diagnostic d'erreur dans le cas où le résultat obtenu ne correspond pas au résultat souhaité.

Si la première intervient évidemment dans toute tâche, la seconde peut être seulement occasionnelle, voire absente : quand les commandes sont exécutables immédiatement, dans beaucoup de cas on ne se préoccupe guère de rechercher l'erreur, d'autant qu'on ne se souvient pas nécessairement de la suite des commandes réalisées, mais plutôt on recommence en changeant éventuellement de procédure. Ce n'est pas le cas, bien sûr, dans les situations de programmation, où le diagnostic d'erreur est une composante fondamentale de la tâche.

Dans la seconde forme d'utilisation, la logique de l'utilisation se confond avec celle du fonctionnement. Il n'est pas question de sous estimer l'importance de la connaissance du fonctionnement, le problème est de savoir quelle est la meilleure voie pour l'acquérir. Vaut-il mieux commencer par donner une connaissance directe en explicitant au maximum l'effet de chaque commande, au besoin en le simulant et en lui donnant une expression visible ? Vaut-il mieux au contraire faire apprendre les utilisations possibles du dispositif, celles qui correspondent à des objectifs d'action que le sujet peut se donner lui-même et reporter la connaissance du fonctionnement dans un second temps en l'introduisant pour justifier le bien fondé des procédures ou pour expliquer que plusieurs procédures puissent être équivalentes ?

4 - Le problème qui demeure dans cette dernière hypothèse est de déterminer quelles sont les bonnes règles d'utilisation à faire apprendre, celles qui correspondent à des actions que le sujet se donne comme objectifs intermédiaires dans la réalisation de ses tâches quand il opère sans l'aide du

dispositif ou qu'il est susceptible de se donner, compte tenu des possibilités nouvelles qui peuvent être offertes par le dispositif. On peut sans doute avancer dans cette voie en analysant la façon dont les opérateurs formulent la programmation de leur activité dans les diverses tâches qu'ils ont à réaliser en examinant si l'on peut trouver dans cette programmation un ensemble d'actions élémentaires que d'une part les sujets formulent (et donc prennent comme sous-but), qui sont communes à un ensemble de tâches assez large et dont le contenu (c'est-à-dire la description en termes plus élémentaires) soit relativement consistant et indépendant du contexte de la tâche. Une autre voie serait d'étudier comment des opérateurs ayant une grande expérience dans l'utilisation d'un dispositif expliquent à un novice comment s'en servir; l'hypothèse de travail est que les experts ont découvert les possibilités de la machine qui sont pertinentes par rapport aux objectifs d'action potentiels, qui fonctionnent comme des macro commandes, et que dans leurs explications ils devraient se référer à des commandes plus élémentaires qui prennent en compte explicitement les contraintes du fonctionnement.

REFERENCES BIBLIOGRAPHIQUES

- ACKERMAN-VALLADO E. (1977). Analyse des procédures de résolutions d'un Problème de composition des hauteurs. Archives de Psychologie, XLV, 101-125.
- ACKERMAN-VALLADO E. (1980). Etudes des relations entre procédures et attribution de significations aux instruments dans une tâche de construction de chemins. Archives de Psychologie, XLV, III, 59-93.
- AUSUBEL D.P. (1968). Educational Psychology, a cognitive view. Holt Rinehart (1968).
- BISSERET A., ENARD C. (1969). Le problème de la structuration de l'apprentissage d'un travail complexe. Bulletin de Psychologie, 23, 11-22, 632-648.
- Du BOULAY B., O'SHEA T., MONK J. (1980). The black box inside the glass box : presenting computing concepts to novices. Int. J. Man. Mach. Studies, 14, 237-249.
- CUNY X., DERANSART P. (1972). Formalisation pour l'analyse du travail mental. Bulletin de Psychologie, 273-281.
- CUNY X., DERANSART P. (1972). Eléments de formalisation pour servir à l'analyse psychologique d'un travail de contrôle. Le Travail Humain, 1, 1-16.
- ENARD C. (1972). Applications de la "Méthode d'Interactions Constantes des Unités Programmées (MICUP) à la formation des contrôleurs de la navigation aérienne. Rapport IRIA C072 11 C17.
- ENARD C., BISSERET A. (1969). L'utilisation de l'enseignement programmé dans la construction de la mémoire permanente chez l'opérateur humain, in La Recherche en Enseignement Programmé. Dunod, 123-216, Paris.
- FISHBEIN E. (1978). Schèmes virtuels et schèmes actifs dans l'apprentissage des sciences. Revue Roumaine des Sciences Sociales, 45, 119-125.

- FRIEMEL E., RICHARD J.F., SILVART L., WEIL-BARAIS A. (1982). Elaboration d'une procédure de résolution compatible avec les contraintes d'un dispositif. Rapport de fin de contrat ADI, Université Paris 8.
- FRIEMEL E., RICHARD J.F., BARAIS A. (1982). Etude de l'apprentissage du fonctionnement d'une calculette. Rapport de fin de contrat ADI, Université de Paris 8.
- GLUCKSBERG S., DANKS J.M. (1969). Functional fixedness : stimulus equivalence mediated by semantic-acoustic similarity. *J. Exper. Psychol.* 74, 400 - 405.
- HALASZ F., MORAN T.P. (1982). Analogy considered harmful. *Human Factors in Computer Systems. Proceeding of the ACM Gainsburg, March 1982*, 383-386.
- HOC J.M. (1978). La programmation informatique comme situation de résolution de problème. Thèse de Doctorat de 3^e cycle Paris 5.
- HOC J.M. (1980). Un exemple de dispositif informatique expérimental pour la psycho-pédagogie de la programmation. *Informatique et Sciences Humaines* 44, 67-86.
- HOC J.M. (1981). Une approche diachronique de la résolution de problème, un exemple : l'apprentissage de la programmation informatique. *Psychologie Française*, 26 (3-4), 182-192.
- INHOLDER B., ACKERMAN-VALLADO E., BLANCHET A., KARMILOFF-SMITH A., KILCHER-HABEDORN A., MONTANGERO J., ROBERT M. (1976). Des structures cognitives aux procédures de découverte. *Archives de Psychologie*, XLIV, 57-71.
- LAURIERE J.L. (1982). Représentation et utilisation des connaissances, les systèmes experts, *Technique et Science Informatique*, 25-42 et 110-133.
- LEMERCIER C. (1982). Analyse différentielle du processus de réalisation lors d'un apprentissage technologique : la conduite de la machine outil. *Psychologie Française*, 27, 113-133.

- LEWIS C., MACK R. (1982). Learning to use a text processing system : evidence from "thinking aloud protocols. Human Factors in Computer Systems. Proceeding of the ACM Gainsburg, March 15-17 1982, 387-392.
- MAYER R.E. (1979). A Psychology of Learning Basic. Communications of the ACM, 22, 11, 589-593.
- MAYER, R.E. (1981). A Psychology of how novices learn computer programming. Computing Survey, 13, 1, 121-141.
- MAYER R.E., BAYMAN P. (1981). Psychology of calculator language : a framework for describing differences in user's knowledge. Communications of the ACM, 24, 8, 511-520.
- MORAN T.P. (1981). The command language grammar : a representation for the user interface of interactive computer systems. Int. J. Man Mach. Studies, 15, 3-50.
- MORE R., RANCOULE Y., LAMOUREUX G. Initiation à l'informatique par une méthode de simulation sur ordinateur. Revue Française de Pédagogie (sous presse).
- NGUYEN-XUAN A. (1982). Le système de production. Revue Française de Pédagogie 60, 331-45.
- PIAGET J. (1974). Réussir et comprendre, Paris P.U.F.
- REUCHLIN M (1973). Formalisation et réalisation dans la pensée naturelle. J. Psychol. Norm. et Path., 4, 389-408.
- PLUM T. (1977). Fooling the user of a programming language. Software Practice and Experience, 7, 215-221.
- SCAPIN D.L. (1982). Conception des langages de commande en langue naturelle restreinte, Thèse de Doctorat de 3° cycle, Paris 5.

SZLICHCINSKI K.P. (1979). Telling people how things work. Applied Ergonomics, 10
1, 2-8

VERMERSCH P. (1976). Une approche de la régulation de l'activité chez l'adulte,
registres de fonctionnement, déséquilibre transitoire et microgénèse.
Thèse de Doctorat de 3^e cycle Paris 5.

WINOGRAD T. (1975). Frame representations and the declarative/procedural controversy,
dans G.BOBROW, D.COLLINS. Ed. Representation and understanding,
studies in cognitive Science, Academic Press, New York, 185-210.

WRIGHT P. (1981). "The instructions clearly state ..." Can't people read ?".
Applied Ergonomics, 12, 3, 131, 141.

YOUNG R.M. (1981). The machine inside the machine : user's models of pocket
calculators, Int. J. Man-Mach. Studies, 15, 51-85.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

